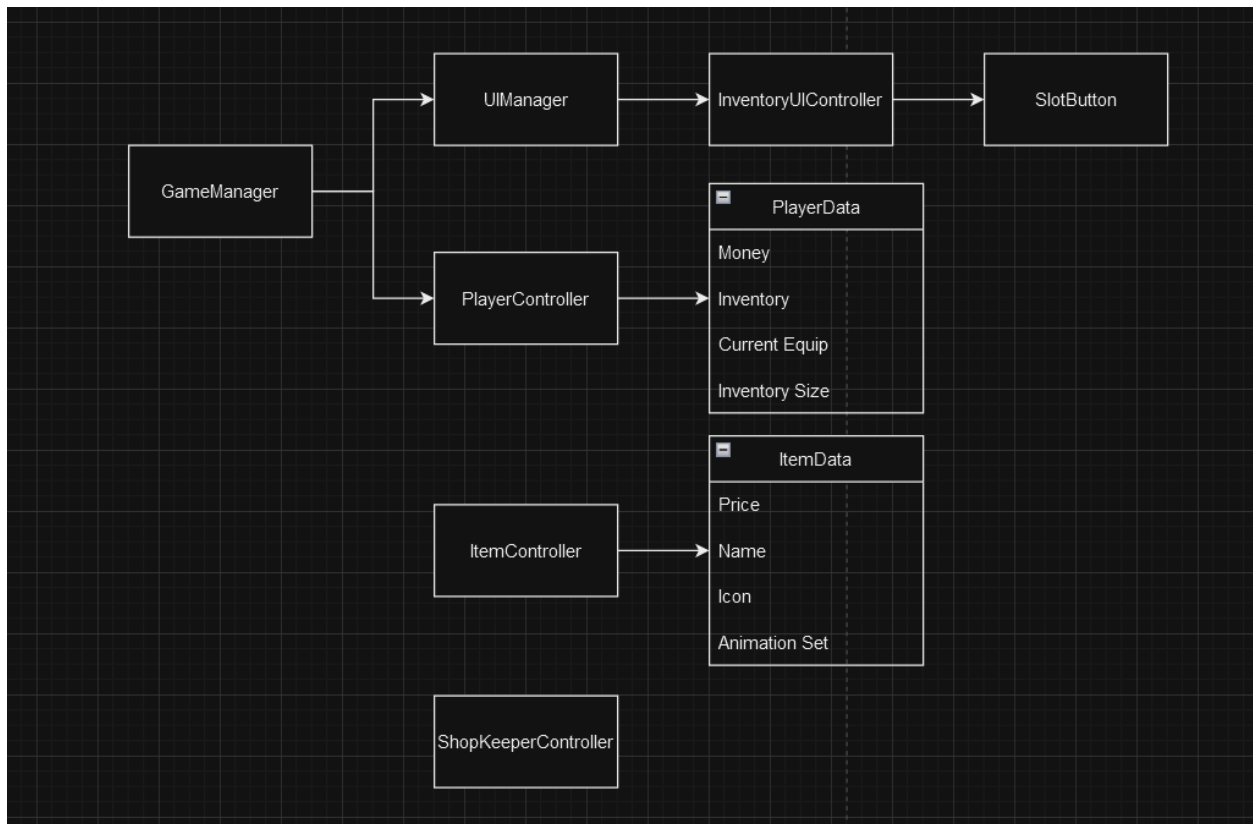


Top Down Shop

Shop Design References



Module Dependency Diagram



1. GameManager:

- The GameManager serves as the central initiator for the Clothing Shop Simulation Game. Responsible for orchestrating the initialization of all modules, it ensures a systematic and orderly setup process. This module plays a crucial role in establishing the foundation for the game environment.

2. Player Controller:

- At the core of player interaction lies the Player Controller module. This component processes player inputs, manages clothing transactions, handles inventory operations, and oversees character animations.

3. Player Data:

- The Player Data module stores essential player information, including currency, inventory content, inventory size limits, and currently equipped clothing. This centralized repository of player-related data streamlines the management of the player's data

4. ItemController:

- The ItemController module manages the initialization logic for in-game items. This includes setting up price tags and acquiring colliders for interaction. By encapsulating item-specific functionalities, ItemController contributes to the modularity and maintainability of the overall system.

5. Item Data:

- Item Data serves as a comprehensive database containing information about each item in the game. This includes item names, prices, animation sets, and icons. The module facilitates the retrieval of essential item details.

6. UIManager:

- The UIManager module takes charge of managing all UI-related modules, paralleling the GameManager's role but specifically for user interface elements. By providing an organized structure for UI components.

7. InventoryUIController:

- As the controller for the inventory menu, the InventoryUIController module handles operations such as equipping and selling items. It acts as the intermediary between player interactions and the player's inventory, enabling seamless customization and management of the player's attire.

8. Slot Button:

- The Slot Button module facilitates the selection of stored items within the inventory. By responding to player inputs, it streamlines the process of interacting with and managing the inventory.

9. ShopKeeperController:

- The ShopKeeperController module contains the logic governing the shopkeeper's behavior, such as displaying the shop icon when the player is in proximity.

Thought Process:

My approach began with a swift analysis of reference games, such as 'Animal Crossing' and 'Stardew Valley,' to outline essential features for the Clothes Shop Simulation Game. Following this, I sought free assets for rotoscope sprite animations, crucial for

outfit changes, streamlining the development process. To enhance visual appeal, I opted for itch.io's free assets for the game's interior, ensuring efficiency in populating the environment without unnecessary asset creation.

Subsequently, I systematically developed required modules, ensuring modularity and ease of maintenance in the project.

Personal Performance Assessment:

Confidence in my engineering capabilities is evident as I promptly initiated development after acquiring suitable assets. This confidence stems from a solid grasp of module requirements and the ability to efficiently translate ideas into code.

Acknowledging the challenge of simultaneous design and coding, I strategically prioritized core mechanics over polish, aligning with an agile development mindset. This approach ensures the robust implementation of essential features before refining the game further.

The decision to separate design and coding processes contributes to a streamlined workflow, allowing dedicated focus on coding once the design aspects are established. While prioritizing core mechanics, I remain open to allocating more time for polishing in subsequent stages of development.

In summary, my thought process emphasizes strategic planning, efficient asset utilization, and a balanced approach between speed and polish.